MapView: Graphical Data Representation for Active Learning

Eva Weigl[1] Alexander Walch[1] Ulrich Neissl[1] Pauline Meyer-Heye[1]
Wolfgang Heidl[1] Thomas Radauer[2] Edwin Lughofer[3] Christian Eitzinger[1]

[1]Profactor GmbH, Steyr-Gleink, Austria
[2]STRATEC Consumables GmbH, Anif, Austria
[3]Johannes Kepler University of Linz, Austria

Abstract

Active learning facilitates the training of classifiers by selectively querying the user in order to gain insights on unlabeled data samples. Until recently, the user had limited abilities to interact with an active learning system: A sub-selection was presented by the system and every sample within had to be annotated. We propose an alternative and graphical solution to active learning called *MapView* where the user may profit from a different interpretation of the underlying data. Experiments underline the usability and advantages of our approach during the training of a classifier from scratch.

Keywords: active learning; graphical data representation; classification; random forests

1 Introduction and Motivation

Active learning is an ongoing research field in the machine learning environment. It enriches and facilitates the training of classifiers by getting rid of enormous amounts of a priori trained samples. This means, instead of forcing a user to manually annotate a large number of samples, the learner pre-selects a subset of *interesting* samples via an algorithm [13]. By presenting these samples to the user, the classifier is able to learn quicker and even explore interpolation or extrapolation areas in the feature space. This avoids tedious annotation times and costs. There are different strategies to select those *interesting* samples [12], with *uncertainty sampling* being the most prominent one which selects the least certain samples according to the currently valid model [13]. Typically, the user then annotates the selected samples, and afterwards the classifier is re-trained or updated based on the newly gained information.

In this paper, we present an alternative approach to user interaction during classifier training within the scope of active learning. Instead of showing the user the sub-set of interesting samples and forcing him/her to annotate all of them, we propose a graphical representation of the data where interesting samples are highlighted within the whole data range. In this way, the user sees all data *at a glance* which results in advantages such as easy identification of clusters as well as outliers.

In the area of *information visualization*, the authors of [1] presented a promising tool for the visualization of probabilistic classification data. A similar technique is presented in [6] or [14]. [2] demonstrate views of clustered graph data. [11] presented a user-based active learning scenario where instead of using classical selection strategies, the user is enabled to choose interesting samples and label them. Our method extends this approach by

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

providing information on the original features in the graphical visualization as well as confidences and proposed class labels from the classifier. This gives the user a more complete picture of the scenario. Also all samples are presented, allowing relabeling of samples and even adding new classes during training.

Our proposed method is independent from the classifier choice as long as the classifier result contains an additional probability measure about how *certain* the classifier is in its decision. We take advantage of a Random Forest classifier [4] and interpret the trees' votes as probabilities. We chose Random Forests because we start with a small number of pre-labeled samples for active learning where a bagged ensemble approach such as Random Forests is well known to outperform other standard (non-ensembled) classifiers. This is because bagging nicely explores the data space through bootstrapping the samples a multiple times, which is, for instance, more deeply analyzed in [3]. In this way, bagging not only reduces the bias of the classifier, but also its variance.

2 Method

We propose a graphical approach to support the user in training a classifier via active learning. The scenario is multi-class classification with pool-based sampling with uncertainty sampling as query strategy.

The main information displayed in the graphical representation is: current labels, current predicted labels, certainty of the prediction and sample coordinates in feature space. The last providing the user with a feature analysis view. This kind of view is described as enhancing the user's understanding of predictions and trust in the underlying classifier in [1]. To display the multi-dimensional feature coordinates, we need to project them onto a 2-dimensional map. For this purpose, we first take advantage of k-means clustering to find a number of cluster centers among our data (corresponding to the number of classes). These centers are then embedded onto a 2-dimensional map by dimensionality reduction. All remaining samples can then be represented as linear combination of the cluster centers and are added to the map. In the following, we will describe the method in more detail.

As classifier, we select Random Forests (see [4] for more information). However, as mentioned before, any classifier that delivers probability measures for each class label can be selected. In our case, we use the votes of each tree divided by the number of all trees as certainty measure for the sampling of interesting samples: the higher the votes, the more certain the classifier is in its decision. This is similar to a least confidence sampling strategy with aspects of maximum entropy sampling in active learning, where the sample to be labeled by the user, is chosen by the learning algorithm [11]. However, in our approach the user is merely given the information as decision support, thus providing an interactive learning scenario as described in [5].

Within the scope of our classifier, we use the term *event* for any item in our database (this can be e.g., an image). Let $\mathbf{E}_i$ be the i-th event in our database, consisting of $i = 1, ..., N$ samples. Each event $\mathbf{E}_i$ has an F-dimensional feature vector $\mathbf{x}_i = \{x_1, ..., x_F\}$. The goal is to assign one of K class labels to the image. For this purpose, for each event $\mathbf{E}_i$ a class vector $\mathbf{c}_i = \{c_1, ..., c_K\}$ exists, with $c_j \in [0; 1], j = \{1, ..., K\}$ representing the probability of belonging to class j.

Since our aim is to embed the high-dimensional events on a user-friendly 2D map, it is necessary to perform dimensionality reduction. Dimensionality reduction means finding an embedding $e : \mathbb{R}^d \to \mathbb{R}^c$ that transforms high-dimensional data from its original d-dimensional space into representations with a reduced number of dimensions c (with $c \ll d$) [15]. In our environment, there are two vectors per event which result in two possibilities for the embedding: (i) the feature vector $\mathbf{x}_i$, or (ii) the class vector $\mathbf{c}_i$. The first has the advantage that the resulting map will not change after each classifier training, since the features remain static. Additionally, the resulting map can be used at the very beginning where not a single class label is present in the dataset. On the other side, using the class vector for the embedding better resembles the classifier's view of the data. Nearby located events are regarded as belonging to a similar class, and uncertainties in labeling can quickly be identified (e.g., events lying between two classes). However, the main drawback is that the map will change after every training or incremental update of the classifier. Since our motivation is to facilitate the annotation work for the user, we chose the feature vectors for our embedding in order to avoid confusion over the continuously changing map as it would be in case of the class vectors.

There exist a variety of dimensionality reduction methods (see [15] for a comparative review). For our experiments, we used the Sammon's Mapping (SM) [10] since it attempts to preserve the inherent data structure. The goal is to minimize the following error function

$$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \tag{1}$$

where d^*_{ij} is the distance between the i-th and j-th event in the original space, and d_{ij} the distance in the embedding. As distance measure the Euclidean distance was selected. The minimization is performed iteratively with a random initialization and a heuristic approach as proposed by [7]. It is a non-linear approach since the embedding is not a linear combination of the original components (as for example, in techniques like the principal component analysis) [8]. In addition, the computational complexity is quadratic (w.r.t. the number of events) which we regarded feasible since the mapping only needs to be computed once as the feature vectors remain static. Additionally, instead of performing the SM on all events from the dataset, we take advantage of k-means clustering as initial step where we compute the locations of K cluster centers (corresponding to the number of K classes). The feature vectors $\mathbf{x_i}$ of all N events of our dataset are then transformed and represented as a linear combination of the K cluster centers $\mathbf{k_j}$ (i.e., as barycentric coordinates):

$$\mathbf{x}_i \approx \sum_{j=1}^{K} \mathbf{k}_j \cdot \lambda_j \tag{2}$$

with $\sum_{j=1}^{K} \lambda_j = 1$. In our application area, the number of cluster centers is typically smaller than the dimension d of the feature vectors. Hence, for each feature vector, we computed a least squares solution of the overdetermined equation system in Eq. 2. In general, the number of cluster centers need to be restricted to $\min(d, K)$ in order to guarantee a unique solution of Eq. 2. By choosing $\lambda_1 = 1 - \sum_{j=2}^{K} \lambda_j$, one can reduce the constraint on the lambdas by solving:

$$\mathbf{x}_i - \mathbf{k}_1 \approx \underbrace{(\mathbf{k}_2 - \mathbf{k}_1, \dots, \mathbf{k}_K - \mathbf{k}_1)}_{\overline{\mathbf{k}}} \cdot \begin{pmatrix} \lambda_2 \\ \vdots \\ \lambda_K \end{pmatrix} \tag{3}$$

which is equivalent to finding a least squares solution of Eq. 2. Since the matrix $\overline{\mathbf{k}}$ is independent of the feature vectors $\mathbf{x}_i$, its pseudoinverse $\overline{\mathbf{k}}^+$ enables an efficient computation of the barycentric coordinates of all feature vectors:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_K \end{pmatrix} = \overline{\mathbf{k}}^+ \cdot (\mathbf{x}_i - \mathbf{k}_0) \tag{4}$$

$$\lambda_1 = 1 - \textstyle\sum_{j=2}^{K} \lambda_j \tag{5}$$

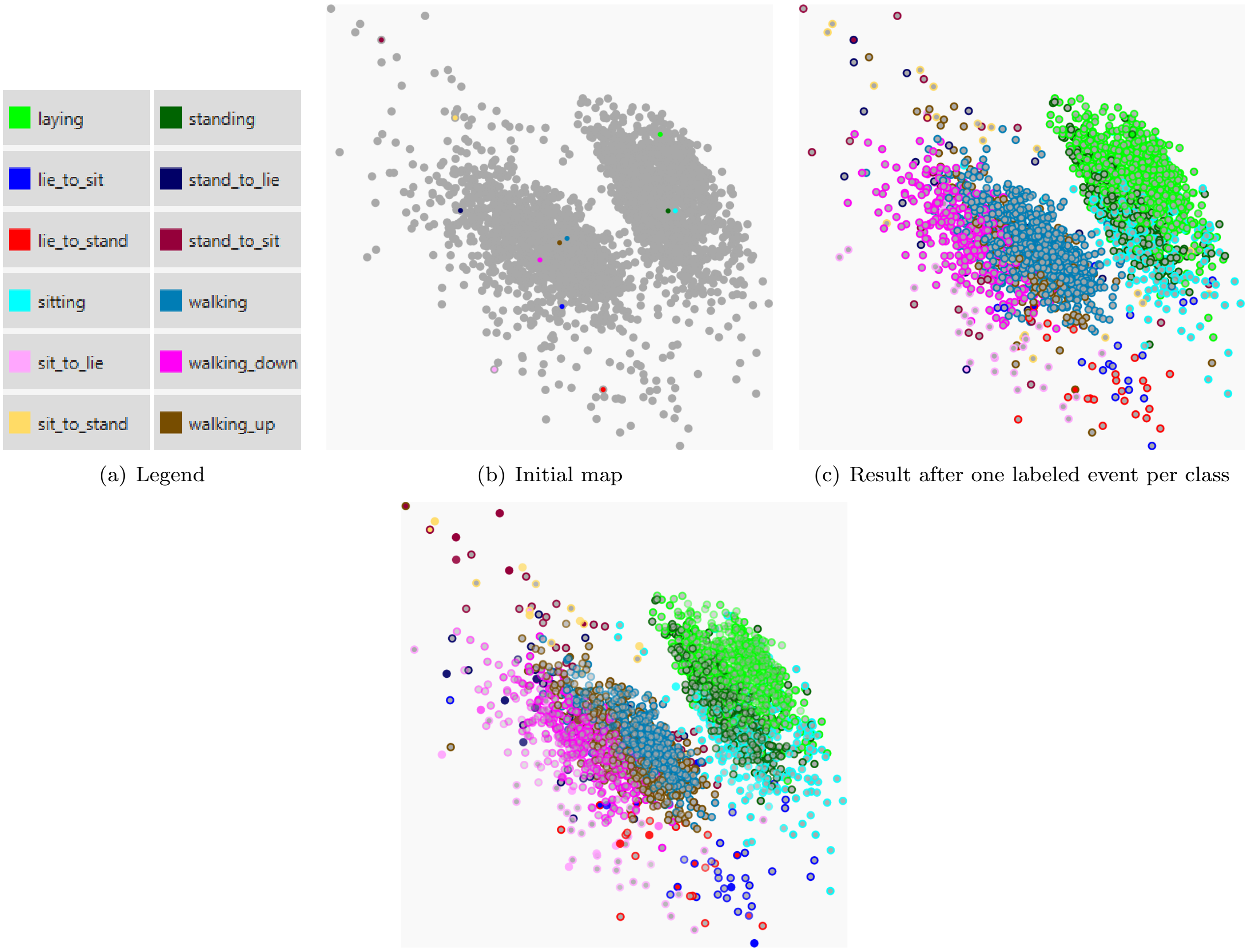
After embedding the cluster centers $\mathbf{k}_j$ via SM, the embedded feature vectors $\mathbf{x}^*_i, i = \{1, ..., N\}$ of all events are computed using Eq. 2 by substituting $\mathbf{k}_j$ with the embedded cluster centers $\mathbf{k}^*_j$.

We implemented a simple user interface depicting the proposed MapView of a given dataset (see Fig. 1). In our setting, all events are displayed as points on a 2-dimensional map. Instead of presenting the user the raw view of a pre-selected set of interesting instances (e.g., unlabeled images with pre-computed features), the user gets a graphical overview. Within this view, events are displayed as points with the transparency of the color proportional to the certainty of the classifier – i.e., the more certain a classifier is about classifying a specific event, the less visible (and thus, more transparent) the corresponding point is on the map. This results in a highlighting of interesting samples that is intended to attract the user's attention, while samples with a high classifier certainty are blurred out. We also decided to show both the manually assigned label (area of the circle) as well as the classifier's prediction (circular ring) pretending the label is unknown. The user can interact with the MapView via different ways: (i) zooming in/out and translating, (ii) getting a preview of a selected event (e.g., image thumbnail and features), (iii) assigning a class label to the event (via a pop-up window that shows a preview of the underlying raw image or features), and (iv) applying a (re-)training of the classifier.

3 Experimental results

For our experiments, we used an activity recognition dataset containing accelerometer and gyroscope sensor signals collected by waist-mounted smartphones (see [9]). The set consists of 12 classes (representing basic activities and postural transitions of human subjects, such as walking, standing, etc.) with a total of 3162 samples with 561 features each.

We attempted to train a classifier *from scratch* by excluding the class labels for our active learning approach and only using them for labeling the query samples. We started with a random labeling of one sample per



(a) Legend (b) Initial map (c) Result after one labeled event per class

(d) Result after 6 labeled events per class

Figure 1: MapView of the dataset. 1(a) Legend of the class labels in the MapView. 1(b) In the beginning, only one event per class is labeled, the remaining events are unlabeled and thus colored gray. 1(c) Result of the classifier training on the initial labeled set (one event per class), and 1(d) after having labeled 5 additional events per class. The more transparent an event occurs, the more confident the classifier is in its decision.

class and a subsequent training of the classifier. This step is depicted in Fig. 1(b) with each color representing a class label (see Fig. 1(a)). The result after training the classifier on these labeled samples is shown in Fig 1(c). Obviously, the certainty of the classifier is very low at this point as nearly all samples are depicted opaque. Afterwards, we labeled five uncertain samples per class and again trained the classifier (the result is in Fig. 1(d)).

The results are easily interpretable and match our assumptions: As one can see, the *active* activities (*walking, walking_down, walking_up*) form a cluster in the lower left region of the map, whereas *passive* activities (*laying, sitting, standing*) are centered on the upper right half of the map, with *laying* being even more separated from the other two classes. Postural transitions (*lie_to_sit, lie_to_stand, sit_to_lie, sit_to_stand, stand_to_lie, stand_to_sit*) are spread in between the other two clusters. The average accuracy as shown in Tab. 1 evaluates the classifier's performance after the initial setup of only one labeled event per class, and after 5 additional samples. We also evaluated the 12 classes by dividing them into passive and active classes as described before, as well as into 3 postural transition classes of the pairs *lie_to_sit/lie_to_stand, sit_to_lie/sit_to_stand, stand_to_lie/stand_to_sit*. Apparently, after labeling only 5 additional events per class, the average accuracy increases up to 60%. When considering the combined groups with only 1 labeled event per (original) class, the accuracy even rises to 97%.

The advantage of our method is that it is faster to calculate the embedding only for the cluster centers instead of all samples of the dataset. In addition, the map also allows a graphical interpretation of the dataset and the classifier's behavior.

Acuracy	1 labeled sample per class	6 labeled samples per class
Per class	0.47	0.60
Per group	0.97	0.98

Table 1: Accuracy of the classification after labeling only one sample per class, and after 5 additionally labeled samples per class. Additionally, we grouped postural transitions as well as active and passive activities and evaluated their average accuracy (last row of table).

4 Conclusion

The proposed method implements a graphical representation of an underlying dataset for classification via active learning. Instead of presenting the user a sub-set of *interesting* samples and requiring their annotation, the samples are depicted as 2D points on a map (called 'MapView') with the color corresponding to the class label. This results in several advantages such as (i) using active learning to avoid the annotation of a large amount of data, (ii) gaining insight into the high-dimensional feature space of the data within the 2D view (e.g., simple identification of cluster samples or outliers), (iii) getting feedback of the classifier's certainty in labeling the samples (samples about which the classifier is uncertain are plotted opaque, whereas the more certain a classifier is, the more transparent the 2D point representation), and (iv) straightforward interpretation of the classifier result and improved understanding of the classifier's decision (e.g., understanding why the classifier made or is uncertain its decision). In the future, we will consider feature selection before the embedding process in order to reduce the high-dimensional space to the most promising features. Additionally, other dimensionality reduction or embedding methods might be worth experimenting with.

Acknowledgements

This work was funded via the projects 'rollerNIL' within the call 'Produktion der Zukunft', and 'MVControl' within the call 'IKT der Zukunft'. Both programs are supplied by the Austrian Research Promotion Agency (FFG), and promoted by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT).

References

[1] Bilal Alsallakh, Allan Hanbury, Helwig Hauser, Silvia Miksch, and Andreas Rauber. Visual methods for analyzing probabilistic classification data. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1703–1712, 2014.

[2] Michael Balzer and Oliver Deussen. Level-of-detail visualization of clustered graph layouts. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 133–140. IEEE, 2007.

[3] P Brazdil, CG Carrier, C Soares, and R Vilalta. *Metalearning: Applications to Data Mining.* Cognitive Technologies. Springer Berlin Heidelberg, 1 edition, 2009.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 23–32. IEEE, 2012.

[6] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[7] T Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences.* Springer Berlin Heidelberg, 2001.

[8] Boaz Lerner, Hugo Guterman, Mayer Aladjem, I Dinsteint, and Y. Romem. On pattern classification with Sammon's nonlinear mapping an experimental study. *Pattern Recognition*, 31(4):371–381, 1998.

[9] J Reyes-Ortiz, Luca Oneto, A Sama, X Parra, and D Anguita. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing*, 171(1):754–767, 2016.

[10] J.W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[11] Christin Seifert and Michael Granitzer. User-Based Active Learning. In *International Conference on Data Mining Workshops*, pages 418–425. IEEE, 2010.

[12] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison, Wisconsin, USA, 2010.

[13] Burr Settles. *Active Learning*, volume 18 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool, San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA), 2012. OCLC: 799360189.

[14] Alexandru Telea and Ozan Ersoy. Image-Based Edge Bundles: Simplified Visualization of Large Graphs. In *Computer Graphics Forum*, volume 29, pages 843–852. Wiley Online Library, 2010.

[15] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: A comparative Review. Technical Report TiCC-TR-2009-005, Tilburg University, The Netherlands, 2009.